UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/647,106	08/25/2003	Andrew James Booker	550-462	9839

23117 7590 12/11/2008

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
CHEN, QING	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
12/11/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/647,106	BOOKER ET AL.	
	Examiner	Art Unit	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 August 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-17,19-29,31-37 and 39-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-17,19-29,31-37 and 39-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *22 August 2008* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment filed on August 22, 2008.

2. **Claims 1-4, 6-17, 19-29, 31-37, and 39-42** are pending.

3. **Claims 1, 7, 9, 14, 20, 22, 27, 32, 34, 35, 41, and 42** have been amended.

4. **Claims 5, 18, 30, and 38** have been canceled.

5. The objection to the drawings is withdrawn in view of Applicant's submission of the replacement drawing sheets.

6. The objections to the abstract are withdrawn in view of Applicant's amendments to the abstract.

7. The objections to Claims 1-4, 6-17, 19-26, and 32 due to an improper antecedent basis and a typographical error are withdrawn in view of Applicant's amendments to the claims. However, the objections to Claims 1-4, 6-17, and 19-26 due to another improper antecedent basis are maintained in view of Applicant's arguments and further explained hereinafter.

8. The 35 U.S.C. § 112, first paragraph, rejections of Claims 14-17, 19-26, 35-37, and 39-42 are withdrawn in view of Applicant's amendments to the claims. However, the 35 U.S.C. § 112, first paragraph, rejections of Claims 27-29 and 31-34 are maintained in view of Applicant's arguments and further explained hereinafter.

9. The 35 U.S.C. § 112, second paragraph, rejections of Claims 1-4, 6-17, 19-29, 31-37, and 39-42 as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention are withdrawn in view of Applicant's arguments or amendments to the claims.

10. The 35 U.S.C. § 112, second paragraph, rejections of Claims 14-17, 19-26, 35-37, and 39-42 as being incomplete for omitting essential steps are withdrawn in view of Applicant's amendments to the claims. However, the 35 U.S.C. § 112, second paragraph, rejections of Claims 27-29 and 31-34 as being incomplete for omitting essential steps are maintained in view of Applicant's arguments and further explained hereinafter.

11. For clarity of the prosecution history record, the Amendment After Final (received on 02/28/2008) was entered and considered by the Examiner. Examiner would like to make clear that the latest Non-Final Rejection (mailed on 03/31/2008) is the fourth Office action for the instant application.

Response to Amendment

Specification

12. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Objections

13. **Claims 1-4, 6-17, and 19-27** are objected to because of the following informalities:

- **Claims 1 and 14** recite the limitation "said sequence of generated instructions." Applicant is advised to change this limitation to read "said corresponding sequence of generated instructions" for the purpose of providing it with proper explicit antecedent basis and/or keeping the claim language consistent throughout the claims.

- **Claims 2-4 and 6-13** depend on Claim 1 and, therefore, suffer the same deficiency as Claim 1.
- **Claims 15-17 and 19-26** depend on Claim 14 and, therefore, suffer the same deficiency as Claim 14.
- **Claim 27** contains a typographical error: A comma (,) should be added between "condition code" and "wherein."

Appropriate correction is required.

Claim Rejections - 35 USC § 112

14. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

15. **Claims 27-29 and 31-34** are rejected under 35 U.S.C. 112, first paragraph, as based on a disclosure which is not enabling. The subject matter of "generat[ing] software test instructions" is considered to be critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976).

The claimed invention omits the critical step of generating software test instructions disclosed to be essential to the invention. The preamble of Claim 27 clearly states that the claim is directed to generating software test instructions. However, the limitations of the claim only

pertain to generating a sequence of instructions from a corresponding sequence of instructions. The claim does not recite any subject matter that relates generating a sequence of instructions with generating software test instructions. Applicant's specification clearly states that the present invention is directed to generating statistical information relating to the operation of software code when tested on a processor *(see page 1, lines 3-6)*. Furthermore, Figure 4B and its corresponding description in the Applicant's specification clearly describe the step of collecting profiling information.

A claim which omits subject matter disclosed to be essential to the invention as described in the specification or in other statements of record may be rejected under 35 U.S.C. 112, first paragraph, as not enabling. See MPEP § 2164.08(c). Such essential matter may include missing elements, steps, or necessary structural cooperative relationships of elements described by the Applicant as necessary to practice the invention.

Claims 28, 29, and 31-34 depend on Claim 27 and, therefore, suffer the same deficiency as Claim 27.

16. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

17. **Claims 27-29 and 31-34** are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01. The omitted step is "generat[ing] software test instructions." The omitted step

is considered to be critical to the claimed invention, since the step is necessary and must occur for the claimed invention to function as intended by the Applicant as described in the disclosure.

Claim Rejections - 35 USC § 102

18. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

19. **Claims 1-4, 6, 10-17, 19, 23-29, 31, 35-37, and 39-42** are rejected under 35 U.S.C. 102(a) as being anticipated by **Applicant Admitted Prior Art (hereinafter "AAPA")**.

As per **Claim 1**, AAPA discloses:

- a) generating, from a sequence of instructions, at least one of which includes a condition code, a corresponding sequence of generated instructions, for selected instructions having a condition code, wherein a corresponding generated instruction is a predetermined generated instruction having a corresponding condition code *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a*

special instruction (SPI) for each instruction in the original opcode 14."; Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code.");

- b) executing, on a target processor, said corresponding sequence of generated instructions and thereby producing software test information *(see Page 6: 10-13, "The original opcode 14 together with the generated opcode 16 is stored in the memory 22. A handler routine 30 is also stored in the memory 22 which is operable by the processor core 20 to generate code coverage and profiling information using the program code 14 and the generated opcode 16.")*; and

- c) when during said step (b) said predetermined generated instruction is encountered, determining with reference to status information associated with an operation of said target processor whether the corresponding condition code of said predetermined generated instruction is satisfied and, if so, replacing said predetermined generated instruction with said corresponding instruction from said sequence of instructions so as to cause said corresponding instruction to be executed, wherein said predetermined generated instruction is an instruction which is not recognised by said target processor *(see Page 6: 14-25, "The processor core 20 retrieves the first instruction of the generated opcode 16. The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction. Once activated, the handler routine 30 refers to the original opcode 14 and checks the condition code of the corresponding original instruction. The handler routine 30 then determines whether the original instruction would have been executed by comparing its condition code with the current status flags of the processor core 20. If the handler routine 30 determines that the original*

instruction would have been executed, then it replaces the special instruction with the original instruction. The handler routine 30 is then exited. The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.").

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- wherein each instruction of said sequence of instructions includes a condition code *(see Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code.").*

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- wherein said condition code is an instruction qualifier which prevents the instruction from being executed by said target processor unless said status information satisfies said condition code *(see Page 2: 5-8, "The condition code indicates the conditions that those flags that must satisfy for the associated instruction to be executed. Such condition codes include EQ/NE (equal/not equal), CS/CC (carry set/carry clear), PL/MI (positive/negative), AL (always), etc.").*

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- wherein said status information is predetermined architectural state associated with said target processor and said condition code specifies a status of said predetermined architectural state that must be met in order for the instruction to be executed *(see Page 2: 8-14, "Hence, by way of example, in the ARM (trademark) instruction set, the condition code EQ/NE*

requires that the zero condition flag (`Z` flag) must be set/cleared respectively for the instruction to be executed; the Z flag is set if the result of the last condition flag setting instruction was zero. Similarly, the condition code PL/MI requires that the negative condition flag (`N` flag) must be cleared/set respectively for the instruction to be executed; the N flag is set if the result of the last condition flag setting instruction was negative.").

As per **Claim 6**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- generating, from said sequence of instructions, a sequence of generated instructions, a predetermined generated instruction being generated for each instruction in the sequence of instructions *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").*

As per **Claim 10**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- incrementing a coverage counter when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original*

instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").

As per **Claim 11**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- incrementing a counter associated with said corresponding instruction when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 12**, the rejection of **Claim 11** is incorporated; and AAPA further discloses:

- replacing a preceding instruction in said sequence of generated instructions with said predetermined generated instruction having a condition code corresponding to said preceding instruction *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 13**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- executing said corresponding instruction on said target processor *(see Page 6: 24 and 25, "The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.")*.

Claims 14-17, 19, and 23-26 are apparatus claims corresponding to the method claims above (Claims 1-4, 6, and 10-13) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1-4, 6, and 10-13.

As per **Claim 27**, AAPA discloses:

- a) generating, from a sequence of instructions, at least one of which includes a condition code, a corresponding sequence of generated instructions, for selected instructions having a condition code, wherein a corresponding generated instruction is a predetermined generated instruction having a corresponding condition code, wherein said predetermined generated instruction is an instruction which is not recognised by a target processor *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14."; Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code."; Page 6: 14-25, "The processor core 20 retrieves the first instruction of the generated opcode 16.*

The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction.").

As per **Claim 28**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- wherein each instruction of said sequence of instructions includes a condition code *(see Page 2: 3 and 4, "The instruction may be conditional and, in which case, may contain a condition code.")*.

As per **Claim 29**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- wherein said condition code is an instruction qualifier which prevents the instruction from being executed by a target processor unless said status information satisfies said condition code *(see Page 2: 5-8, "The condition code indicates the conditions that those flags that must satisfy for the associated instruction to be executed. Such condition codes include EQ/NE (equal/not equal), CS/CC (carry set/carry clear), PL/MI (positive/negative), AL (always), etc.")*.

As per **Claim 31**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- generating, from said sequence of instructions, a sequence of generated instructions, a predetermined generated instruction being generated for each instruction in the sequence of instructions *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the*

program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").

As per **Claim 35**, AAPA discloses:

- a) producing software test information by executing, on a target processor, a sequence of generated instructions *(see Page 6: 10-13, "The original opcode 14 together with the generated opcode 16 is stored in the memory 22. A handler routine 30 is also stored in the memory 22 which is operable by the processor core 20 to generate code coverage and profiling information using the program code 14 and the generated opcode 16.")*; and

- b) when a predetermined generated instruction is encountered during said step (a), determining with reference to status information associated with an operation of said target processor whether a condition code of that predetermined generated instruction is satisfied and, if so, replacing said predetermined generated instruction with a corresponding instruction from a sequence of instructions so as to cause said corresponding instruction to be executed, wherein said predetermined generated instruction is an instruction which is not recognised by said target processor *(see Page 6: 14-25, "The processor core 20 retrieves the first instruction of the generated opcode 16. The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction. Once activated, the handler routine 30 refers to the original opcode 14 and checks the condition code of the corresponding*

original instruction. The handler routine 30 then determines whether the original instruction would have been executed by comparing its condition code with the current status flags of the processor core 20. If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction. The handler routine 30 is then exited. The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.").

As per **Claim 36**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- wherein said condition code is an instruction qualifier which prevents the instruction from being executed by said target processor unless said status information satisfies said condition code *(see Page 2: 5-8, "The condition code indicates the conditions that those flags that must satisfy for the associated instruction to be executed. Such condition codes include EQ/NE (equal/not equal), CS/CC (carry set/carry clear), PL/MI (positive/negative), AL (always), etc.").*

As per **Claim 37**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- wherein said status information is predetermined architectural state associated with said target processor and said condition code specifies a status of said predetermined architectural state that must be met in order for the instruction to be executed *(see Page 2: 8-14, "Hence, by way of example, in the ARM (trademark) instruction set, the condition code EQ/NE requires that the zero condition flag (`Z` flag) must be set/cleared respectively for the instruction to be executed; the Z flag is set if the result of the last condition flag setting instruction was zero.*

Similarly, the condition code PL/MI requires that the negative condition flag (`N` flag) must be cleared/set respectively for the instruction to be executed; the N flag is set if the result of the last condition flag setting instruction was negative.").

As per **Claim 39**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- incrementing a coverage counter when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 40**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- incrementing a counter associated with said corresponding instruction when the condition code of the predetermined generated instruction is satisfied to provide an indication that said corresponding instruction will be executed *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.").*

As per **Claim 41**, the rejection of **Claim 40** is incorporated; and AAPA further discloses:

- replacing a preceding instruction in a sequence of generated instructions with said predetermined generated instruction having a condition code corresponding to said preceding instruction *(see Page 7: 10-14, "If the handler routine 30 determines that the original instruction would have been executed, then it replaces the special instruction with the original instruction, increments a counter associated with that instruction and replaces the preceding instruction with a special instruction.")*.

As per **Claim 42**, the rejection of **Claim 35** is incorporated; and AAPA further discloses:

- executing said corresponding instruction on said target processor *(see Page 6: 24 and 25, "The processor core 20 hardware can then execute the original instruction which has just replaced the special instruction.")*.

Claim Rejections - 35 USC § 103

20. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

21. **Claims 7-9, 20-22, and 32-34** are rejected under 35 U.S.C. 103(a) as being unpatentable over **AAPA** in view of **US 5,712,996 (hereinafter "Schepers")**.

As per **Claim 7**, the rejection of **Claim 1** is incorporated; and AAPA further discloses:

- a2) generating said predetermined generated instruction for one instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").*

However, AAPA does not disclose:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions.

Schepers discloses:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions *(see Abstract, "In order to be able to execute rapid processing of a program on super-scalar microprocessors, the individual instructions of this program must be divided into instruction groups, which can be processed by processing units of the microprocessor, in such a way that the instructions can be processed in parallel.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions. The modification would be obvious

because one of ordinary skill in the art would be motivated to execute rapid processing of the program code *(see Schepers – Abstract)*.

As per **Claim 8**, the rejection of **Claim 7** is incorporated; and AAPA further discloses:

- generating said predetermined generated instruction for the last instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.")*.

As per **Claim 9**, the rejection of **Claim 7** is incorporated; however, AAPA does not disclose:

- wherein said predetermined generated instruction provides information relating to the number of instructions in a corresponding instruction group of said number of instruction groups.

Schepers discloses:

- wherein said predetermined generated instruction provides information relating to the number of instructions in a corresponding instruction group of said number of instruction groups *(see Column 2: 63-65, "The number of the components per instruction group is fixed in*

accordance with the number of the instructions which a microprocessor can load simultaneously.").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include wherein said predetermined generated instruction provides information relating to the number of instructions in a corresponding instruction group of said number of instruction groups. The modification would be obvious because one of ordinary skill in the art would be motivated to determine the number of instructions a microprocessor can load simultaneously *(see Schepers – Column 2: 63-65)*.

Claim 20 is rejected for the same reason set forth in the rejection of Claim 7.

Claim 21 is rejected for the same reason set forth in the rejection of Claim 8.

Claim 22 is rejected for the same reason set forth in the rejection of Claim 9.

As per **Claim 32**, the rejection of **Claim 27** is incorporated; and AAPA further discloses:

- a2) generating said predetermined generated instruction for one instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated*

opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").

However, AAPA does not disclose:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions.

Schepers discloses:

- a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions *(see Abstract, "In order to be able to execute rapid processing of a program on super-scalar microprocessors, the individual instructions of this program must be divided into instruction groups, which can be processed by processing units of the microprocessor, in such a way that the instructions can be processed in parallel.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include a1) partitioning said sequence of instructions into a number of instruction groups, each instruction group including one or more instructions. The modification would be obvious because one of ordinary skill in the art would be motivated to execute rapid processing of the program code *(see Schepers – Abstract).*

As per **Claim 33**, the rejection of **Claim 32** is incorporated; and AAPA further discloses:

- generating said predetermined generated instruction for the last instruction in each of said instruction groups *(see Page 1: 22-28, "An instruction set emulator 12, which is a software*

program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14.").

As per **Claim 34**, the rejection of **Claim 32** is incorporated; however, AAPA does not disclose:

- wherein said predetermined generated instruction provides information relating to the number of instructions in a corresponding instruction group of said number of instruction groups.

Schepers discloses:

- wherein said predetermined generated instruction provides information relating to the number of instructions in a corresponding instruction group of said number of instruction groups *(see Column 2: 63-65, "The number of the components per instruction group is fixed in accordance with the number of the instructions which a microprocessor can load simultaneously.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Schepers into the teaching of AAPA to include wherein said predetermined generated instruction provides information relating to the number of instructions in a corresponding instruction group of said number of instruction groups. The modification would be obvious because one of ordinary skill in the art would be motivated

to determine the number of instructions a microprocessor can load simultaneously *(see Schepers – Column 2: 63-65)*.

Response to Arguments

22. Applicant's arguments filed on August 22, 2008 have been fully considered, but they are not persuasive.

In the Remarks, Applicant argues:

a) Claims 1 and 14 specify at line 4 "a corresponding sequence of generated instructions" and it is noted that this phrase in each claim provides literal antecedent basis for the term "instructions," the phrase "generated instructions," the phrase "sequence of generated instructions" and the phrase "a corresponding sequence of generated instructions" as long as there are no other "instructions" specified or "generated instructions" specified or "sequence of generated instructions" specified. A review of Applicants' claim 1 shows there to be no other "sequence of generated instructions" other than the sequence introduced at line 4 and therefore the subsequent reference to "said sequence of generated instructions" has explicit literal antecedent basis in claim 1, as well as claim 14 and any further objection is respectfully traversed.

Examiner's response:

a) Examiner disagrees. Applicant's arguments are not persuasive for at least the following reasons:

First, with respect to the Applicant's assertion that the subsequent reference to "said sequence of generated instructions" has explicit literal antecedent basis, the Examiner respectfully submits that the limitation "said sequence of generated instructions" does not have proper explicit antecedent basis. The claims recite two groups of instructions. The first group is the original sequence of instructions and the second group is the corresponding sequence of instructions generated from the original sequence of instructions. Thus, there is an established relationship between the original sequence of instructions and the generated sequence of instructions. Presumably, the limitation "said sequence of generated instructions" refers to the corresponding sequence of instructions generated from the original sequence of instructions. The claim language has to set forth this established relationship for the claims to be in proper form.

Second, Examiner respectfully submits that providing the limitations of the claims with proper explicit antecedent basis would improve the clarity or precision of the claim language used or, at the very least, keep the claim language consistent throughout the claims. Examiner further submits the relevant portions of MPEP § 2173.02 and 37 CFR 1.75 with emphasis added for purposes of convenience in discussion and illustration:

> **MPEP § 2173.02 Clarity and Precision**
> The examiner's focus during examination of claims for compliance with the requirement for definiteness of 35 U.S.C. 112, second paragraph, is whether the claim meets the threshold requirements of clarity and precision, not whether more suitable language or modes of expression are available. When the examiner is satisfied that patentable subject matter is disclosed, and it is apparent to the examiner that the claims are directed to such patentable subject matter, he or she should allow claims which define the patentable subject matter with a reasonable degree of particularity and distinctness. Some latitude in the manner of expression and the aptness of terms should be permitted even though the claim language is not as precise as the examiner might desire. **Examiners are encouraged to suggest claim language to applicants to improve the clarity or precision of the**

> **language used**, but should not reject claims or insist on their own preferences if other modes of expression selected by applicants satisfy the statutory requirement.
>
> **37 CFR 1.75 Claim(s).**
> (a) The specification must conclude with a claim **particularly pointing out** and **distinctly claiming** the subject matter which the applicant regards as his invention or discovery.

According to the section of the MPEP and the patent rule provided above, the Examiner would like to point out that a claim must particularly point out and distinctly claim the subject matter which the Applicant regards as the invention. In accordance with MPEP § 2173.02, the Examiner suggests providing the limitations of the claims with proper explicit antecedent basis and/or keeping the claim language consistent throughout the claims to improve the clarity or precision of the claim language used. Hence, doing so would help the Examiner in reviewing the claims for compliance with 35 U.S.C. § 112, second paragraph.

Therefore, for at least the reasons set forth above, the objections to Claims 1-4, 6-17, and 19-26 are proper and therefore, maintained.

In the Remarks, Applicant argues:

b) However, with respect to claim 27, Applicants believe it inappropriate to specify the step of generating software test information because the claim itself is a method of generating software test instructions. The claim literally specifies the steps of how those software test instructions are generated, i.e., "generating, from a sequence of instructions, at least one of which includes a condition code, a corresponding sequence of generated instructions, for selected instructions having a condition code ..." Thus, the method by which "software test instructions"

are generated is set out in the steps of the claim. Accordingly, claim 27 as it currently exists is believed to be correct and reconsideration of the § 112 (first paragraph) rejection of this claim and claims dependent thereon is respectfully traversed.

Examiner's response:

b) Examiner disagrees. With respect to the Applicant's assertion that the method by which "software test instructions" are generated is set out in the steps of the claim, the Examiner respectfully submits that the step performed in Claim 27 only specifies generating, from a sequence of instructions, a corresponding sequence of generated instructions. Thus, one of ordinary skill in the art would not readily recognize that a corresponding sequence of generated instructions, at least on the *prima facie* level, is a sequence of software test instructions. Furthermore, as evident by independent Claims 1, 14, and 35, the Applicant has set forth that the software test information are produced by executing the corresponding sequence of generated instructions on a target processor. Claim 27 does not recite or suggest, either explicitly or implicitly, such feature of the other independent claims.

Therefore, for at least the reason set forth above, the rejections made under 35 U.S.C. § 112, first and second paragraphs, with respect to Claims 27-29 and 31-34 are proper and therefore, maintained.

In the Remarks, Applicant argues:

c) An examination of Applicants' specification at page 1, lines 22-28, does not indicate any disclosure of a "predetermined generated instruction having a corresponding condition code"

and, in fact, merely discloses that in the AAPA, a previously known instruction set emulator produces generated opcode and the analysis module substitutes a special instruction for each instruction in the original opcode. There is no indication that a condition code is associated with the original opcode. The Examiner also references the specification pages 2, 3 and 4, but apparently fails to appreciate that the condition codes which are referenced in the specification are all condition codes associated with the original opcodes and not the generated opcodes (it is noted that claim 1 specifies a predetermined generated instruction).

Examiner's response:

c) Examiner disagrees. Applicant's arguments are not persuasive for at least the following reasons:

First, with respect to the Applicant's assertion that there is no indication that a condition code is associated with the original opcode, the Examiner respectfully submits that the Applicant later acknowledges, in the same argument paragraph, that AAPA describes the condition codes being associated with the original opcodes.

Second, with respect to the Applicant's assertion that the condition codes are associated with the original opcodes and not the generated opcodes, the Examiner respectfully submits that AAPA clearly discloses "wherein a corresponding generated instruction being a predetermined generated instruction having a corresponding condition code" *(see Page 1: 22-28, "An instruction set emulator 12, which is a software program which models the operation of a particular predetermined processor, is loaded onto the computing device 10. Also loaded onto the computing device 10 is the original opcode 14 of the program code to be analysed. From this*

original opcode 14, an analysis module of the instruction set emulator 12 produces generated opcode 16. In order to produce the generated opcode 16, the analysis module substitutes a special instruction (SPI) for each instruction in the original opcode 14."). Note that in order to produce the generated opcode, a special instruction (a corresponding condition code) is substituted for each instruction in the original opcode.

Third, the claims recite only "corresponding condition code" with no further clarification on the claim scope of the limitation "corresponding condition code" as intended by the Applicant to cover. The claims are not limited to the scope of the condition code and the corresponding condition code being the same condition code. Thus, as the claims are interpreted as broadly as their terms reasonably allow (see MPEP § 2111.01 I), the interpretation of a broad limitation of "corresponding condition code" as a special instruction and the like by one of ordinary skill in the art is considered to be reasonable by its plain meaning.

Therefore, for at least the reasons set forth above, the rejections made under 35 U.S.C. § 102(a) with respect to Claims 1, 14, 27, and 35 are proper and therefore, maintained.

In the Remarks, Applicant argues:

d) In fact, line 18 of the specification states, and it is quoted in the outstanding Official Action (at page 9 beginning at line 16), "once activated, the handler routine 30 refers to the original opcode 14 and checks the condition code of the corresponding original instruction." This language very clearly specifies exactly what is done in all of the prior art, i.e., the computer has to go back and check to see whether or not a condition code is satisfied and clearly leads one away from the present invention in which there is provided a corresponding condition code with

the generated instruction which clearly reduces the time to determine whether that condition code is satisfied because it avoids the need to invoke a software handler routine to go back and check the original instruction. Thus, the Examiner's own quotation illustrates that, not only does the AAPA fail to anticipate Applicants' independent claims or claims dependent thereon, it actually teaches away from the claimed invention by suggesting that "once activated, the handler routine 30 refers to the original opcode 14 and checks the condition code of the corresponding original instruction."

Examiner's response:

d) Examiner disagrees. Applicant's arguments are not persuasive for at least the following reasons:

First, with respect to the Applicant's assertion that AAPA fails to disclose "determining with reference to status information associated with an operation of said target processor whether the corresponding condition code of said predetermined generated instruction is satisfied," the Examiner respectfully submits that AAPA clearly discloses "determining with reference to status information associated with an operation of said target processor whether the corresponding condition code of said predetermined generated instruction is satisfied" *(see Page 6: 14-17, "The processor core 20 retrieves the first instruction of the generated opcode 16. The processor core 20 determines whether that instruction is a special instruction and, if so, then the handler routine 30 is invoked; otherwise, the processor core 20 executes the instruction and then retrieves the next instruction.")*. Note that when an instruction of the generated opcode is retrieved, it is

checked to determine whether it is a special instruction (whether the corresponding condition code of said predetermined generated instruction is satisfied).

Second, with respect to the Applicant's assertion that AAPA teaches away from the claimed invention, the Examiner respectfully submits MPEP § 2131.05 with emphasis added for purposes of convenience in discussion and illustration:

> **MPEP § 2131.05 Nonanalogous >or Disparaging Prior< Art**
> "**Arguments that the alleged anticipatory prior art** is nonanalogous art' or **teaches away from the invention**' or is not recognized as solving the problem solved by the claimed invention, **[are] not germane' to a rejection under section 102.**" *Twin Disc, Inc. v. United States*, 231 USPQ 417, 424 (Cl. Ct. 1986) (quoting *In re Self*, 671 F.2d 1344, 213 USPQ 1, 7 (CCPA 1982)). See also *State Contracting & Eng' g Corp. v. Condotte America, Inc.*, 346 F.3d 1057, 1068, 68 USPQ2d 1481, 1488 (Fed. Cir. 2003) (The question of whether a reference is analogous art is not relevant to whether that reference anticipates. **A reference may be directed to an entirely different problem than the one addressed by the inventor, or may be from an entirely different field of endeavor than that of the claimed invention, yet the reference is still anticipatory if it explicitly or inherently discloses every limitation recited in the claims.**).
>
> **A reference is no less anticipatory if, after disclosing the invention, the reference then disparages it. The question whether a reference "teaches away" from the invention is inapplicable to an anticipation analysis.** *Celeritas Technologies Ltd. v. Rockwell International Corp.*, 150 F.3d 1354, 1361, 47 USPQ2d 1516, 1522-23 (Fed. Cir. 1998) (The prior art was held to anticipate the claims even though it taught away from the claimed invention. "The fact that a modem with a single carrier data signal is shown to be less than optimal does not vitiate the fact that it is disclosed."). >See *Upsher-Smith Labs. v. Pamlab, LLC*, 412 F.3d 1319, 1323, 75 USPQ2d 1213, 1215 (Fed. Cir. 2005)(claimed composition that expressly excluded an ingredient held anticipated by reference composition that optionally included that same ingredient);< see also *Atlas Powder Co. v. IRECO, Inc.*, 190 F.3d 1342, 1349, 51 USPQ2d 1943, 1948 (Fed. Cir. 1999) (Claimed composition was anticipated by prior art reference that inherently met claim limitation of "sufficient aeration" even though reference taught away from air entrapment or purposeful aeration.).

According to the section of the MPEP provided above, the Examiner would like to point out that arguments pertaining to a reference "teaching away" from the claimed invention are inapplicable to an anticipation analysis. In the instant application, AAPA describes prior art systems of generating software test information relating to the operation of software code when tested on a target processor. Due to the broad language of the claims, the prior art is deemed to anticipate the claims even though it teaches away from the claimed invention.

Therefore, for at least the reasons set forth above, the rejections made under 35 U.S.C. § 102(a) with respect to Claims 1, 14, 27, and 35 are proper and therefore, maintained.

Conclusion

23. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

24. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Q. C./

Examiner, Art Unit 2191

/Wei Y Zhen/

Supervisory Patent Examiner, Art Unit 2191